1 Concepts and Bases of Information Security
- 绝对安全的系统是不可能达到的
- 黑客不追求刺激或挑战，他们为了某些“利益”而进行黑客攻击
- 超过80%的攻击来自内部，由于内部员工更清楚怎么从外部攻击本系统
- 内部用户(insider)是对环境或系统的某些方面有合法的接触或联系的人；相比外部攻击者，内部用户有更多的机会和知识；内部用户通常有明确的动机
- 有报导的黑客事件比实际发生的少得多
- 太公兵法(阴符，一本书分成三个 part)是中国的第一个密码(1st cipher in china)
- 豪密是中华人民共和国的第一个密码(1st cipher in CCP)
- 凯撒密码(Caesar cipher)是已知的最早的替换式密码(substitution cipher)，有 25 种可能的密钥(secret keys)(移位数)
- 棒子密码(scytale cipher)，用同样粗细的棒子解密，密钥是棒子的粗细
- 信息隐写术(information hiding/Steganography)
  - 写进头皮、蜡烛里、鸡蛋里、微点(microdot)与电影、培根密码(bacon cipher)使用字体加密
- 密码学演化(evolution)
  - 第一次：**科托夫原则**(kerckhoff principle)，密码学从经验转向科学
    - 增强安全性：凡是长时间使用并不改变的东西，都认为是公开信息
    - 加密算法不具有保密性，安全性只依赖于于密钥的保密
    - 密码的强度必须在对手已知算法的前提下进行定义
    - 使量产密码机成为可能
  - 第二次：电子计算机，密码学从人工转向机器化和电子化
    - 计算机出现在二战中，为了破解 ENIGMA
    - 计算机加速了加密和解密
    - 理论基础：香农信息论(Shannon Information Theory)
    - 现代密码学的出现：DES
  - 第三次：公钥密码学(public key cipher)，密码机制(cryptography mechanism)的革命化演变
    - 密码学的新方向
    - 公钥密码学允许双方在不分享任何密钥的情况下交换大量信息
    - 最广泛使用的公钥密码算法：RSA
  - 第四次：因特网科技和应用派生的密码学内涵和外延的拓展
    - 密码学内涵(connotation)拓展：不只是秘密，还有其他问题
    - 密码学延伸(extension)的拓展
      - 应用：军事->民用->个人
      - 保护：简单的 PTP 协议->复杂的 PTP 协议->多种网络协议和应用->5G、物联网、人工智能
      - 软硬件的发展带来的挑战：硬件能力的提升和开销的降低；分布式计算、云计算、密码分析
- 基于计算机的信息安全和基于纸笔的信息安全
  - 原件和复印件的可区分性
  - 修改(alteration)是否留下痕迹
  - 易销毁性：纸相对较难销毁
  - 手签字基于物理特征；数字签名基于二进制信息
- 计算机安全的特征
  - 全面性(comprehensive)：一个系统安全取决于最弱的一环
  - 过程性(procedural)：不断来会上升的螺旋式上升安全模型
  - 动态性(dynamic)：整个安全系统处在不断的进步、更新过程
  - 层次性(hierarchy)：必须使用多层安全技术去解决安全风险
  - 相对性(relative)：安全是相对的，没有绝对安全
- 计算机安全的三元素
  - **保密性**(Confidentiality)：别人是否能看到我们的数据
  - **完整性**(Integrity)：我们的数据是否会被非法修改
  - **可用性**(Availablity)：资源是否可用
  > - 真实性(Authenticity)：信息的来源是否是声明的来源
- 计算机安全概念
  - 资产(Asset)：人、财产、信息
  - 威胁(Threat)：任何可能有意或无意利用漏洞并获得，损坏或破坏资产的东西
  - 漏洞(Vulnerable)：安全程序中的弱点或漏洞可被威胁利用以获取对资产的未授权访问
  - 风险(Risk)：由于利用漏洞引起的威胁而造成资产损失，损坏或破坏的可能性
- 安全威胁类型：
  - 自然威胁：地震火灾洪水闪电
  - 物理威胁：错误使用、不小心损坏
  - 硬件/软件威胁：不合适的设计、后门、逻辑炸弹、系统冲突
  - 介质威胁：硬盘(HDD)损坏、不小心删除、不小心退磁
  - 泄露(leak)威胁：电磁泄露，屏幕监视
  - 通信威胁：抓包、通信过程中的修改和伪造
  - 个人威胁：不小心删除，故意破坏和泄露
- 安全攻击类型：
  - **中断**(Interruptiom)：使系统毁坏或不可用；攻击“可用性”；易检测
    - 中断方式：硬件损坏、传输链损坏、引入噪声、摧毁路由、DOS 攻击
  - **截获**(Interception)：未授权用户获取对数据的访问；攻击“保密性”；难以检测
    - 截获方式：窃听(eavesdropping)、链路监控(link monitoring)、抓包、黑入　- 无法完全避免
  - **修改**(Modification)：未授权用户获取对数据的访问和篡改；攻击“完整性”
    - 修改方式：修改数据库记录、黑入、通信延迟、修改硬件
    - 通过数字水印和其他技术可以避免
  - **伪造**(Fabrication)：未授权用户向系统中插入虚假实体并伪装自己为授权用户；攻击“真实性”
    - 伪造方式：向数据库插入记录、伪造 IP、钓鱼
    - 与不可拒(non-repudiation)相关
- 安全攻击类型：主动/被动攻击
  - **被动攻击**：目标是截获；难以探测，预防比探测更有效
  - **主动攻击**：目标是中断、修改和伪造；易探测，难以预防，可以从攻击中恢复
    - 举例：拒绝服务(DOS)攻击、重放(Replay)攻击
- 对抗安全威胁的目标
  - 阻止(Prevention)：阻止攻击者违反安全策略
  - 检测(Detection)：检测攻击者违反安全策略的情况
  - 恢复(Recovery)：攻击被阻止，系统被修复，恢复操作；即使受到攻击仍然正常运行
- 对抗安全威胁的安全服务
  - 认证(authentication)：确保通信实体是认证的实体，包括对等实体认证和数据来源认证
  - 访问控制(access control)：阻止未认证用户访问
  - 数据保密性：防止数据窃听
  - 数据完整性：确保数据发送到认证用户，未经修改等
  - 不可拒：阻值来自任何实体的通信拒绝
  - 可用性：保证服务的可用性
- 对抗安全威胁三成靠技术七成靠管理，人的问题是迄今为止安全问题的主要来源

2 Cryptography
- 明文、密文、密钥、加密解密算法的关系：$C=E_k(P),P=D_k(C)$
- **替换**(Substitution)和**位移**(Transposition)
- 古典密码->一战->机械密码->二战后->计算机密码
- 古典密码：棍子密码、古希腊密码(二维密码表　字母->二维坐标)、凯撒密码、女皇玛丽密码(字母->符号)、维吉尼亚密码(vigenere square)、书密码
- 机械密码
  - 齿轮机：ENIGMA
    - 多表替代的加密算法
    - 加密解密算法是机器的物理结构、齿轮(rotor)
    - 密钥是齿轮的初始设置(齿轮的排列顺序、每个齿轮的初始状态)、接线板的设置
    - 满足科托夫原则
  - ENIGMA 破解：(间谍)获取机器；(滥用)day key 和 message key 的误用，重复导致模式(pattern)的出现；Bombe 机
  - 破解加密算法的途径：寻找模式、减少复杂度、爆破(brute-force)、数学方法和技巧
- 计算机密码
  - 对称(Symmetric)密码学和非对称(Asymmetric)密码学
  - **对称密钥加密算法**(Symmetric Key Ciphers)/共享(Shared)密钥加密算法/保密(Secure)密钥加密算法：加密解密算法相同且使用相同密钥
  - 块加密(Block Cipher)：长 bit 流->数个短 bit 节，加密每一个节，节与节之间的加密没有相互依赖；一个好的块加密算法，输出应该是输入的 bit 和密钥的函数
  - 费斯托密码结构(Feistel Cipher Structure)
    - 目前现代对称密码算法都基于这一结构
    - 使用块加密，增加了块大小
    - 使用两种方法来达成雪崩效应(Avalanche Effect)，即当输入发生微小改变也会导　致输出的不可区分性改变(输出中每个二进制位有50%的概率发生反转)
    - 扩散(Diffusion)：使密文的统计特性与明文之间的关系尽量复杂。迭代交换左右　的一半
    - 扰乱(Confusion)：使密文的统计特性与密钥之间的关系尽量复杂。块大小、密钥　长度、循环轮数、子密钥生成算法、轮函数 F
  - DES(Data Encryption Standard)算法
    - 块大小为 64bits，密钥长度为 56bits
    - 特点：雪崩效应强；很强的反破解能力，只能爆破；如今 56bits 密钥已经不够安全
    - 三重 DES
    - $C = E_{K_3}\left(D_{K_2}\left(E_{K_1}(P)\right)\right)$
    - 向前兼容：当 K2\==K3 || K1==K2 时，等价于 DES 算法
    - 密钥长度为 56*3
    - PGP 和 S/MIME 采用的算法
  - AES 算法：块大小为 128bits，密钥长度为 128/192/256bits；免疫已知的所有攻击；全平台计算快；设计简单
  - 操作模式(Mode of Operation)——块与块之间的处理方式
    - ECB：将长明文直接拆开转成密文再合并，P1 为 1-to-1 变成 C1,但是会被重放攻击(通过 Ck 猜出 Pk) 雪崩效应小，易被重放攻击；ECB 对于同样的明文块会被加密成相同的密文块；因此，它不能很好的隐藏数据模式
    - CBC：将前一个加密的块同时影响后续的加密，造成雪崩效应，上一个块的结果会影响下一个块
  - 流加密(Stream Cipher)：用随机 bit 序列和 P 异或得到 C
  - 密钥分配问题
    - 根据科托夫原则，密钥十分重要，如何共享？
    - 对称密码学的密钥共享方法：
      - A 选择一个密钥并物理传送给 B

    - 第三方选择一个密钥并物理传送给 A 和 B
    - A 和 B 协商更换新密钥
    - 第三方帮助 A 和 B 协商更换新密钥
  - 通常的解决方法 KDC(Key Distribution Center)
  - 对称密码学存在的问题
    - A 加密文件后如何把密码告诉 B
    - 用邮件告诉 B，那么邮件被截获了怎么办
    - B 改了半天交给 A 文件，A 怎么知道真的是 B 改的
- **公钥密码**学/非对称密码学
  - 公钥密码学基于数学函数而不是替换和位移
  - 公钥密码学是非对称的，用两个独立的密钥，也叫做非对称密码学
  - 公钥密码学允许双方在交换秘密信息时不共享任何密钥——解决了密钥分配问题
  - 公钥密码学保证秘密通信——解决了数字签名问题
- 秘密通信：用 A 的公钥加密，发送给 A，A 用 A 的私钥解密
- 认证：B 用 B 的私钥加密文件，发给 A，A 用 B 的公钥验证
- 公钥 KU(Public Key)和私钥 KR(Private Key)的区别
  - 攻击者无法从公钥得到私钥
  - A 和 B 不需要提前共享私钥
  - 无论有多少通信需要完成，只需要一对公钥和私钥
  - 公钥密码学的关键是寻找一个单向函数要满足，计算结果十分容易，但是反向计算不可能
- 公钥密码学的使用：解密/加密；数字签名；密钥交换
- 公钥密码学的要求：密钥对生成容易；加密快解密快(合法情况内)；攻击者通过公钥无法计算私钥；攻击者无法通过密文和公钥得到明文；密钥对可以更改
- 公钥密码学的发展：DH 密钥交换算法、RSA 算法、EIGamal 算法、Elliptic Curves 算法
- **Diffie-Hellman 算法**
  - 用作密钥协商，不是公开密钥加密算法
  - g 是 p 的原根(primitive root)，满足条件 $g \bmod p, g^2 \bmod p, \cdots, g^{p-1} \bmod p$ 是不同 1 到 p-1 的不同整数
  - 给定任意 1 到 p-1 之间的整数 A,存在唯一的指数 a 使得 $A = g^a \bmod p$，a 称为 A 的离散对数(discrete logarithm)
  - 给定 p 和 g，如果 p 非常大，由 a 计算 A 很容易，由 A 计算 a 很难只能穷举
  - $K = A^b \bmod p = (g^a \bmod p)^b \bmod p = g^{ab} \bmod p = (g^b \bmod p)^a \bmod p = B^a \bmod p$
  - a, b, p 的选择将很大程度影响算法的安全性，如果 p 很小，结果很容易被爆破
  - 通常 p 至少是 300 位的质数，a 和 b 至少 100 位，g 为 2/3/5
  - DH 算法没有认证环节，很容易受到中间人攻击
- **RSA 算法**
  - 单向函数：大质数乘法(large primes multiplication)很容易，但是大合数的质因数分解(factorization)十分困难
  - 欧拉函数(Euler's totient function) $\phi(n)$：比 n 小且与 n 互质(coprime to)的正整数的个数
  - 如果 n 是质数，$\phi(n) = n - 1$；如果 n 是合数(composite number)，可以被分解为 $n = \Pi p_i$，则 $\phi(n) = n(1 - 1/p_1)(1 - 1/p_2)\cdots(1 - 1/p_k)$
  - 如果 p 和 q 互质，则 $\phi(pq) = \phi(p)\phi(q)$
  - Euler Theorem:如果 n 和 a 互质 $a^{\phi(n)} \equiv 1 (\bmod n)$
  - Fermat Little Theorem:如果 p 是质数，对任意正整数 a 有 $a^p \equiv a (\bmod p)$;如果 a 不能被 p 整除，$a^{p-1} \equiv 1 (\bmod n)$
  - 密钥对生成和解密加密过程
    - B 生成密钥对，把公钥发送给 A
      - 选择两个大质数(至少 100 位) p 和 q 相乘得到 n
      - 选择两个数 e, d 满足：e 和 (p-1)(q-1) 互质，并且 e 和 d 小于(p-1)(q-1)；e*d 模(p-1)(q-1)余 1
      - 公开(e，n)作为公钥，保留 d 作为私钥
    - A 加密明文 m 变成 c(m 必须小于 n)，发送给 B：找到 B 的公钥(e，n)，计算 $c = m^e \bmod n$；把密文 c 发送给 B
    - B 收到 C，解密：使用私钥计算 $m = c^d \bmod n$
  - RSA 算法的安全性
    - 攻击者可以得到 c 和(e，n)，想获得 m 有以下途径
      - 得到 B 的私钥 d
      - m 是 1 到 n 的数字，可以爆破
      - 尝试从(e，n)计算出私钥 d：一旦 n 可以被分解成 p 和 q，d 就很好找到了
      - 攻击者生成自己的密钥对：攻击者冒充 B 告诉 A 自己的新公钥，此后攻击者就可以用自己的私钥解密了(数字签名解决)
    - 对称密码学和非对称密码学的比较
      - 对称密码学的优点是快且便宜，缺点是密钥分配问题
      - 非对称密码学的优点是密钥分配问题被解决，缺点是贵且慢
      - 在生产生活中使用公钥密码学来进行密钥交换(RSA)；使用对称密码学来进行加密解密(AES)
      - 常见两个误解：公钥加密在防范密码攻击上比常规密码更安全，但实际上二者都依赖于密钥长度和解密的计算量，二者不用比；公钥加密使得常规加密过时，但实际上公钥加密开销相对较大，公钥加密更多的用于密钥管理和数字签名而非直接的加密解密

3 Security Services

- 密码学可以进行 AB 之间的秘密通讯，但加密解密不能防止欺骗(deceiving)，无法做身份认证
- 手写签名的特点：可信；不能被伪造；不能重用；签名文档不可修改；不能否认；签名和时间在某些情况下需要绑定(bound)；要有合法证明(legal previdence)
- 数字签名的特点：可以与签名文档绑定；接收者可以验证签名，其他人无法伪造签名；签名者不可否认签名；签名必须有第三方确认以解决争端；作者、时间戳都是签名的内容
- 数字签名的要求：依赖于内容以防止修改；使用发送者独有的信息以防止否认和伪造；认证和校验比较简单
- 数字签名算法
  - 公钥密码学可以支持数字签名：DSS/DSA; RSA; Elliptic curve
  - 公钥密码学签名过程：A 用私钥签名文档发送给 B，B 用 A 的公钥验证签名
  - 公钥密码学直接签名的问题：文档很长的时候，签名也特别长
- 数字签名应该使用单向哈希函数
  - 任意长度的二进制字符串均转化为固定长度的二进制字符串，可以计算出文档的哈希值并签名
  - 常用的单项哈希函数：MD5，128bits；SHA，160bits
- 单项哈希函数算法的特点：
  - 易计算，给定一个文件，可以很快算出来哈希值
  - 难以反向计算，给定哈希值，算不出源文件
  - 难以寻找碰撞，很难寻找到两个哈希值相同的文件
- 单项哈希函数的安全
  - 原象(Preimage)攻击：给定哈希值，找到原文件
  - 第二原象(second-preimage)攻击：给定文件，找出另一个与原文件哈希值相同的文件
  - 碰撞(Collision)攻击：寻找两个哈希值相同的文件
- 大文件的签名过程
  - 源文件通过单项哈希函数得到哈希值
  - 哈希值通过 A 的私钥加密成为签名
  - 签名和明文文件传送给 B
  - B 用 A 的公钥解密签名得到哈希值 1
  - B 用同样的哈希函数计算明文文件的哈希值 2
  - 对比哈希值 1 和 2 是否相同
- MAC(Message Authentication Code)消息验证码，带密钥的哈希函数
  - 消息完整性服务，用来确认信息传输过程中是否被修改过
- MAC 和数字签名的区别
  - MAC 收发双方需要共享一个秘密 MAC 算法的密钥；MAC 不提供不可否认服务：所有可以验证消息保密性的人都能生成一个 MAC
  - 数字签名收发双方不需要共享任何秘密信息；数字签名可以被有公钥的任何接收者确认；数字签名提供不可否认服务
- MAC 的使用原因：MAC 比数字签名更快；不可否认服务不总是必须的
- PGP 被广泛使用在数字签名、完整性校验、信息加密、数据压缩、邮件格式兼容；多平台兼容；基于密码安全算法
- 认证(Authentication)的实现
  - 你有什么：身份证、密码、密钥
  - 你知道什么：密码、生日、预设问题的答案
  - 你在哪：IP 地址
  - 你是谁：生物认证(指纹、掌印)、行为认证(手写、行走)
- 基于密码的认证
  - 最广泛应用——系统保存认证用户和密码，存在的缺陷：密码在用户和系统间传递时很容易发生窃听漏洞(vulnerable to eavesdropping)；密码文件很难保密；容易记忆的密码很容易猜出来
  - 影响密码系统安全性的因素和机制：输入密码的频率/次数限制；储存密码的方式，哈希后较为安全；网络传输过程的内容形式，哈希后较为安全；修改密码的过程；密码的生命周期；使用*代替密码，不在屏幕显示；定义密码的长度和格式；一段时间后重新输入密码；多次错误后禁止登陆
  - 密码实现：史前明文存密码；早期哈希后存密码；早些时候的 unix 密码使用 DES 作为哈希函数，但是储存在一个所有用户都可以读的文件里，有很大的隐患，文件/etc/passwd
  - 字典攻击：计算出字典里每一个哈希值，然后去对照密码文件里有没有这个哈希值，可行性是来源于很多密码都是来自于一个小字典的
    - 改进方法 1——加盐(Salting)：向用户密码添加随机的盐，再做哈希储存起来；没加盐时候，一轮字典就能搞定所有密码条目；加盐后，一轮字典只能搞定一个密码条目；攻击者必须对一个密码条目尝试全部的盐
  - 改进方法 2——隐密码(Shadow Passwords)：哈希密码不储存在之前的文件里，在 etc/shadow 文件里，只有 root 用户可以访问；添加了密码的有效日期
  - 保护密码的方法：采用数字+字符，使用随机密码；密码安全性检查；Trojan protection，使用安全登陆工具；不同的网站上使用不同的站点，可以同类网站使用相同的密码；不要相信任何第三方的记住密码软件；不要在公共场所和不可信的网路环境下登陆重要账号
  - 其他改进方式：Biometric/Behaviometric Authentication
- 生物认证
  - 通过物理特征进行认证，例如指纹、面部识别、DNA、虹膜(Iris)扫描、手形
  - 优点是不会被盗窃、忘记或丢失
  - 缺点是设备和维护的开销、对比的准确性以及隐私问题
  - 生物认证的错误率
    - False Accept Rate：接收了恶意(malicious)用户
    - False Reject Rate：拒绝了正规用户
    - 增加接受阈值会增加错误接受率并降低错误拒绝率
    - 如果错误接受率和错误拒绝率相等，叫做等错误率(equal error rate)
    - 不同的情况下对二者的概率有不同的要求
  - 生物信息的风险
    - 信息记录时可能不符合要求

    - 可以通过使用留存的记录进行攻击，比如别处搞来的旧指纹
- 网络认证
  - Protocol v1.0:直接声明
  - Protocol v2.0:声明+IP 地址
  - Protocol v3.0:声明+IP 地址+密码
  - Protocol v4.0:声明+IP 地址+加密过的密码
  - 解决重放攻击的方法：挑战响应认证(Challenge Response Authentication)，挑战码(Challenge Code)只被使用一次
  - Protocol v5.0:B 发送任意挑战码 R 给 A，A 须用共享密钥加密回复 R
  - Protocol v6.0:使用挑战码和公钥
- KERBEROS
  - 要解决的问题：在分布式情况下，客户希望在服务器上获得服务。 服务器可以限制授权用户的访问，并可以区分所需的服务
  - 目标：安全、可靠、透明、规模（可扩容）
  - 实现：提供一个认证服务器来识别服务器和使用者；基于传统加密，不使用公钥加密
  - 基本概念：C(client)，AS(Authentication Server)，TGS(Ticket Granting Server)，V(Application Server)，IDc(User’s Identifier on C)，IDv(Identifier of V)，Pc(Password on C)，ADc(Network Address of C)，Kv(shared key of AS and V)
  - KERBEROS v4 基础版：引入 AS 存储所用用户密码，AS 与每个服务器分享一个密钥，C 访问 AS 得到访问 V 的门票并访问 V；问题：需要经常性输入密码、密码明文传输、票一次泄露就可以导致伪装 C 来进行的重放攻击
  - KERBEROS v4 改进版：引入 TGS，C 访问 AS 得到 TGS 的门票，再访问 TGS，由 TGS 给访问 V 的门票，其中两张门票有 lifetime；改进：减少密码输入频率，防止重放攻击，不明文传输密码。Liftime 设置的长短非常重要，太长太短都不好。V 也不好证明自己
- V4 最终版：C 访问 TGS 和 V 时出示自己的认证信息，交流过程中信息与 Kc，IDc，ADc 等有关；依赖于一个安全的时间服务(用到大量时间戳)；需要一个可信的第三方
  - Kerberos V4 与 V5 对比：V4 依赖于 DES，V5 可以自选其他算法；V4 只在 IP 协议上有效，V5 可以用于其他网络地址；V4 的 liftime 上限为 1280min，V5 没有限制；V5 允许身份认证传输
  - Kerberos 缺点：依赖时间、guessing attack、密码存储
- 授权(Authorization)的概念
  - 授权和访问控制(Access Control)基本是一个意思
  - 认证提供基本的授权；认证提供基本的校验用户身份的功能；授权在更深的控制中被需要
  - 授权的三要素：
    - 主体(subject)：可以访问客体的实体，必须用户和进程之类的
    - 对象(object)：可以被访问的东西，比如文件、程序、数据
    - 权限(privilege)：主体访问客体的方法，读、写、删除、运行、子权限(sublicense)
  - 安全访问控制
    - 主要功能：授权(Authorization)，撤销(Revoke)，检查(Checker)
    - 两个方面：制定政策(make policy)和执行政策(execute policy)
- 安全访问控制模型
- DAC, Discretionary Access Control 自主访问控制
  - 特性：根据主体的身份和访问权限做出决定;自主选择(Discretionary)意味着具有某种特权的主体可以自动将其访问权的子集授予其他主体
  - 缺陷：访问允许可能会在信息传输过程中发生改变
  - 访问控制矩阵
    - 访问控制表(access control list)：每个对象添加一个详细列表以访问其主体
    - 能力表(capability list)：每个主体添加一个详细列表以访问其对象
  - Unix OS 使用访问控制列表管理文件
- MAC, Mandatory Access Control 强制访问控制
  - 主要应用于军事(Military)相关领域
  - 主题和对象被标记为不同安全等级：Top Secret/Secret/Confidential/Unclassified
  - Bell-LaPadula 模型保证保密性
    - No Read Up：主体只能读同级或低级的对象
    - No Write Down：主体只能写高级或同级的对象
  - Biba Model 保证完整性，与 Bell-LaPadula 完全相反
- 隐通道(Covert Channel)：使用系统资源进行通信而不会注意到主体(进程)的通道
  - 资源耗尽型隐通道(Resource Exhaustion Channel)：用低级别进程申请资源成功与否来判断高层进程是否运行
  - 负载感知型隐通道(Load Sensing Channel)：用低级别进程执行需要高级别进程所需资源的任务，观察其完成速度确定高级别进程的运行情况
  - 应对隐通道的方法：关闭或者降低通道速度；检测企图使用隐通道的进程；忍受他的存在
  - 隐通道无法完全被避免：限制资源共享，例如资源共享仅可以发生在相同安全级别的进程之间；限制带宽(bandwidth)
- RBAC, Role-Based Access Control 基于角色的访问控制
  - DAC 的安全性太弱，无法满足需求，而 MAC 太强，不够灵活；DAC 和 MAC 的管理工作量都很大;RBAC 是现代信息企业出于安全信息管理的需求，主要为了减少管理人员的工作量
  - RBAC 给每个用户授予一系列不同的角色
  - 特性：职责分离；角色分层；角色激活；用户成员资格约束
  - RBAC 包含五个基本数据客体：用户(users)；角色(roles)；客体(objects)；操作(operations)；许可(permissions)
  - 角色：一组用户的每个角色都与相关操作相关，用户所属角色有权执行这些操作
    - 角色和组的区别：组是一系列用户，角色是一群用户+一群操作许可
  - 关系：多对多，用户被赋予特定的角色，角色被赋予特定的许可
  - 会话(Sessions)：匹配用户和激活的角色
- RBAC 的优势
  - 授权管理很容易
  - 根据工作需要促进分类，如通过财务人员的角色区分公司财务部门和非财务部门员工的公司财务许可
  - 轻松实现最小权限，即使用户被分配为高级身份，也只有在必要时才有权限，以防止事故发生
  - 便于任务分享，不同的角色执行不同的任务
  - 易于进行文件分级管理，文件本身可以分为不同的角色，由用户的不同角色所拥有
  - RBAC 可以用来实现 DAC 和 MAC
- 安全访问控制的原则
  - 模型是固定的，但是策略是灵活的，适当的安全策略是安全的核心
  - 授权管理
    - MAC：允许访问(Allowed access)完全基于主体和对象的安全级别，主体的安全级别由安全系统管理员赋予，对象的安全级别由创造他的主体所决定
    - DAC：多种授权管理方法
      - 集中管理，管理员授权和删除访问控制授权
      - 分级管理，中央管理员将管理职责分配给实际进行集中管理的其他管理员
      - 所有权管理，对象的所有者授权和未经授权的对象访问者
      - 协同管理，特殊资源的访问不是由单个用户决定的，而是由共享用户的合作授权决定的。Collaborative
      - 分布式管理，在分布式管理中，对象的所有者可以授权其他用户进行管理的权限
    - RBAC：RBAC 提供许多类似于自我管理策略的访问控制。 但是，管理权限的授权是 RBAC 的一个重要特性，DAC 和 MAC 中不存在。
  - 缩小用户权限——参照权力分立，将超级用户原有的权利分配给三种特权用户
    - 系统管理员负责系统维护，用户管理，软件安装等功能
    - 安全管理员负责安全规则配置，安全策略管理和其他功能
    - 审计管理员负责检查审计记录，监视系统安全性和其他功能
  - 每个特权用户彼此履行职责和约束。基本约束是，在一类用户眼中，其他类用户拥有的权限与普通用户完全相同。 没有特别的权利。
- 安全访问控制的核心原则
  - 最小权限——即使用户被分配为高级身份，也只有在必要时才有权限，以防止事故发生
  - 权力分立，将超级用户原有的权利分配给三种特权用户
    - 系统管理员负责系统维护，用户管理，软件安装等功能
    - 安全管理员负责安全规则配置，安全策略管理和其他功能
    - 审计管理员负责检查审计记录，监视系统安全性和其他功能
  - 每个特权用户彼此履行职责和约束。基本约束是在一类用户眼中，其他类用户拥有的权限与普通用户完全相同。没有特别的权利

4 Internet and Security

- 网络结构
  - TCP/IP 协议栈进行路由和连接
  - BGP(Border Gateway Protocol)进行路由器发现
  - DNS(Domain Name System)进行域名解析 IP
- TCP 协议栈
  - 数据链路层(ARP, PPP,…)——帧
  - 网络层(IP, ICMP,…)——包
  - 传输层(TCP, UDP)——段
  - 应用层(HTTP, FTP, DNS,…)——信息
- IP 和 TCP
  - IP 负责传送
  - TCP 负责源到目的的可靠性和流控制
    - 滑动窗口——流控制
    - 序列号和确认(acknowledgments)——可靠性
    - 同步(synchronization)(建立虚拟连接)
- IP 和 UDP(User Datagram Protocol)
  - IP 提供路由，IP 地址从某台特定机器获取数据报
  - UDP 用端口区分传输
    - 目的端口号区分数据对应的应用进程
    - 源端口提供返回地址
  - 最小保证：没有确认；没有流控制；没有消息继续(message continuation)
- TCP(Transmission Control Protocol)
  - 发送者：把数据分散成段
    - 每个包都绑定上序列号
  - 接收者：重新组织各个段到正确的顺序
    - Acknowledge receipt；丢包会被重新发送
  - 双方都保持连接状态
  - 面向连接的可靠的协议，通过提供滑动窗口来提供流控制，通过提供序列号和确认来保证可靠性
  - TCP 重发所有没收到的包并且支持终端用户的虚拟连接
  - TCP 的优点是提供了段的确认接收
- 三次握手
  - 序列号用于跟踪包的序列号同时确保没有丢包
  - ISNs 初始序列号是开始号，在 TCP 连接建立的时候使用
- TCP 常见端口号——21：FTP，53：DNS，80：HTTP，443：HTTPS

- DNS 服务
  - DNS 根服务器负责最高级域名解析
    - 当本地 DNS 挂掉，他会找他的授权服务器去问
    - 如果一直挂，通过级联询问一直到根服务器
- 窃听(Sniffing)
  - 许多传输是不加密的：ftp 和 telnet 明文传输密码；许多 web 应用使用 http，没有加密
  - 混杂模式(Promiscuous mode)网络接口卡可以读取所有数据
- ARP 欺骗(Spoofing)
  - ARP 欺骗的运作原理是由攻击者发送假的 ARP 数据包到网络上，尤其是送到网关上。其目的是要让送至特定的 IP 地址的流量被错误送到攻击者所取代的地方
  - ARP：IP->MAC，RARP：MAC->IP
  - 操作系统通过 ARP 缓存来实现 ARP
  - 可以利用 ARP 欺骗实现截获攻击、中间人攻击、DOS 攻击
- IP 欺骗：带有假的源 IP 地址的 IP 协议分组(数据报)，目的是隐藏发送方或冒充另一个计算系统身份
- TCP SYN Flooding
  - 原理：攻击者伪造 IP 地址发送大量请求，被攻击的主机为每个资源分配资源，一旦资源耗尽，客户端就无法正常连接了
  - 这是经典的 DOS 攻击：发送者不需要耗费资源，但是接收者需要对每个请求创建一个线程；非对称的攻击
  - 防止 TCP flooding 的方法
    - 随机删除：如果 SYN 队列满了，随机删除一个
      - 因为正常的连接会很快结束，而且就算被删除，也会重新发起访问，所以不会有很大影响
    - SYN Cookies
      - 可行原因：非对称的资源分配
      - SYN cookie 使得服务器不会储存状态，除非从一个客户端收到了至少两次消息
        - 服务器把 sockets information(IP，端口和服务器和客户端)存在 cookie 里，然后把 cookie 发回客户端
        - 客户端必须把这个 cookie 再发回到服务器做二次请求，然后服务器重新计算 cookie 校验
  - TCP SYN Predition Attack
    - 序列号如果被攻击者预测，那么就能伪造包发给接收方，按照伪造的序列号拼出来满足攻击者目的的包
    - 是很多其它攻击的源:TCP 欺骗、TCP 连接劫持(hijacking)、TCP 重置(reset)
    - 对抗方法：选择一个随机的 initial SYN(ISN)去抵抗预测
  - TCP 阻塞(Congestion)控制
    - 丢包意味着网络阻塞，TCP 协议需要发送方
      - 降低一半速度直到不发生丢包或者速度降到 0
      - 如果丢包停止，传输速度将缓慢增加
    - 解决办法：增加 ack nonces，在 ack 时返回 nonce 来证明他不是一个欺骗
  - DNS 欺骗：修改 DNS 服务器或者本地 DNS 服务器，一般是修改 DNS 缓存数据库，重新定向到页面请求到错误的 IP 地址，导致访问到了错误的服务器
  - 对抗 DNS 欺骗，DNSSEC SEC 指系统安全拓展
    - DNS 欺骗发生的原因：DNS 请求和响应都是未经认证的，导致攻击者可以伪造 DNS 信息
    - DNSSEC：所有的响应都是认证的；既不用于提供加密服务，也不用来面对 DOS 攻击
  - IPSEC
    - IPSEC 支持在 IP 层的所有网络传输的加密和认证
    - IPV6 必须支持 IPSEC，IPV4 是可以选择的
    - 三个核心部分：
      - 验证头(AH, Authentication Headers)：为 IP 包提供数据完整性和认证服务
      - 载荷安全性封装(ESP, Encapsulating Security)：提供安全性、保密性以及认证服务
      - 安全关联(SA, Security Associations)：用来集成安全服务，SA 定义了一系列的算法和常数(密钥等等)对一个发送-接收流中的加密和认证；如果需要双向安全通讯，需要用两个 SA；一个 SA 由三个常量唯一确定：SPI(security parameter index), IP 目的地址,安全协议标识，确定是 AH 还是 ESP；SA 是一组逻辑安全参数，可以将信息共享给另一个实体
    - IPSEC 操作模式
      - IPSEC 可以用在 P2P 或者网络隧道(tunnel)传输
      - 传输模式：只保护 IP 包内容，而不保护 IP 头；由于 IP 头未被修改,所以路由过程不会受到影响。传输层和应用层的数据都受到保护；一般用在 P2P 传输
      - 隧道模式：会加密整个 IP 包，原始 IP 数据包将被隐藏到一个新的 IP 数据包中，并且将附加一个新的 IP 标头；通常用于保护网络和网络之间的 VPN，主机到网络通信以及点对点通信
    - IPSEC 的优点
      - IPSEC 可以增强和实现防火墙/路由器：所有通过边界的数据包都将得到安全性增强；受防火墙保护的主机不需要处理安全问题
      - IPSEC 对最终用户是透明的：构建在 IPSEC 网络上的应用程序无需做任何特殊的事情；自动保证保密性和完整性
    - SSL&TLS 概念
      - SSL 连接(SSL connection)
        - 连接是提供适当类型服务的传输(OSI 层定义)
        - SSL 连接是一种点对点关系。 连接是临时的，每个连接都与一个会话相关联
      - SSL 会话(SSL session)
        - SSL 会话是客户端和服务器之间的关联。 会话由握手协议创建。会话定义由一组连接共享的密码安全参数
        - 避免昂贵的谈判价格来提供每个连接安全参数。
    - SSL&TLS 协议栈
      - SSL&TLS 处在传输层和应用层之间，自身也被分为两层
        - 握手层，定义了三个子协议，Handshake sub protocol, Change Cipher Spec sub protocol, Alert sub protocol
        - 记录层，接收并加密来自应用层的信息，然后发给传输层
      - 握手层——SSL 中最复杂的部分
        - 使服务器和客户端相互验证
        - 协商加密算法，MAC 算法和加密密钥
        - 在应用程序数据传输之前执行握手协议
        - 过程：建立安全谈判，服务器认证和密钥交换，客户端认证和密钥交换，结束
      - 记录层过程：碎片化(Fragmentation)，压缩(可选)，MAC 计算，加密，加入 SSL 记录头
    - 恶意代码是导致违反网站安全策略的一组指令
    - 木马：同时具有公开(overt)目的和隐蔽(covert)目的的程序
      - 木马的复制(replicating)是一个很大的特点，而且很难检测
    - 病毒：将自己插入一个或多个文件并执行一些操作的程序
      - 插入阶段必须存在，不必总是执行，只有启动文件未被感染时，病毒才会将其自身插入启动文件
      - 可以通过跳转等方式将恶意代码藏在执行文件内以躲过杀毒软件的病毒库匹配
      - 病毒也可以加密，通过一段解密代码放在加密代码前面，以躲过侦测(侦测该病毒方法：侦测解密段(正常程序不会加密自己)) Encryted Viruses 加密病毒
      - 通过改变加密方式或者多次加密的方式可以让病毒发生"变异"Polymorphic Viruses 多态病毒
      - 病毒还可以通过增加无用代码的方式来绕过杀毒软件的检查
      - 杀毒软件,除了检测代码匹配,还会检查动作,比如修改注册表,IE 浏览器默认主页地址
    - 蠕虫
      - 蠕虫、木马、病毒的区别
        - 病毒通常插入到宿主代码中，不是一个独立存在的程序，通过感染其他文件进行传播
        - 木马是单独的可运行程序，不用自我复制
        - 蠕虫是单独的可运行程序，自我复制或到其他目的系统自动传播
      - 没有算法可以检测全部的恶意代码
    - 僵尸网络：能够按照指令行事的自治程序网络；通常是一大组可以远程控制的僵尸系统；机器的主人不知道他们已经被控制了；通过 IRC 或者 P2P 进行控制和升级
      - 作为多种攻击的平台：DDOS；垃圾邮件和点击欺诈 spam and click fraud；为新的蠕虫做准备
    - DOS 攻击目的：压倒受害者机器并拒绝为其合法客户提供服务
    - DDOS 攻击
      - 建立一个僵尸僵尸网络
        - 多层架构：使用一些僵尸作为“主人”来控制其他僵尸
      - 命令僵尸对受害者发起协同攻击
        - 不需要欺骗，因为没有必要为了保护一台僵尸机器来使用欺骗，比起伪造 IP 地址，可能获取一台新的僵尸机要快得多
        - 即使在发生 SYN 泛滥的情况下，SYN cookies 也无济于事因为 DDOS 引发的 SYN flood 不同于 DOS,每台僵尸机都是在正常访问服务器的，不会被 SYN cookie 影响
      - 通过来自不同来源的数千台机器的流量抵达受害者
    - 僵尸网络的检测
      - 僵尸机通过 IRC 和 DNS 进行检测
        - IRC 网络聊天室
        - 僵尸使用 DNS 查找主人，并由主人查找僵尸是否被列入黑名单
      - IRC 和 DNS 活动在网络上十分明显：寻找执行扫描的主机，以及拥有高比例主机的 IRC 频道；查找请求多次 DNS 查询的主机，但几乎不会收到有关他们自己的查询
      - 通过使用加密和 P2P 轻松避开检测
    - 邮件欺骗
      - 邮件通过 SMTP 服务进行传送，SMTP 服务没有认证服务
      - 邮件来自 这一项是由发件人设置的
        - 不正确输入验证的经典示例
      - 收件人的邮件服务器只能看到其收到信息的直接对等方的 IP 地址
    - 为什么要隐藏垃圾邮件的源
      - 许多邮件系统会拉黑那些发送大量垃圾邮件的服务器和 ISP(互联网供应商)
      - 黑名单会减少 15-25%的垃圾邮件
      - 垃圾邮件发送者的目的：避免黑名单
        - 僵尸网络来的很容易
    - 垃圾邮件是从哪里来的
      - 垃圾邮件源在整个网络上广泛分布
      - 绝大多数垃圾邮件来自 IP 地址空间的一小部分
      - 垃圾邮件制造者使用基础路由设备
    - 垃圾邮件的对抗：政策性对抗、科技型对抗(过滤器；发件人校验；质询认证)